

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

UNCLASSIFIED

**EXECUTIVE SUMMARY
NUMERICAL AERODYNAMIC SIMULATION FACILITY
FEASIBILITY STUDY**

March 1979

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

Prepared under Contract No. NAS2-9897 by

**Burroughs Corporation
Paoli, Pa.**

for

**AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**

UNCLASSIFIED

INTRODUCTION

This executive summary presents the results of Burroughs Corporation's efforts on the Feasibility Study for the Numerical Aerodynamic Simulation Facility (NASF). The study has demonstrated that a particular form and architecture for the NASF (proposed originally during the Preliminary Study [1, 2] and improved during the present study) would meet the established objectives. A detailed report [3] describes the many facets of the work including the hardware configuration, software, user language, fault tolerance, and other aspects of the system on which this demonstration of feasibility is based.

The Numerical Aerodynamic Simulation Facility is conceived to be more than just a very high-speed computing machine. The facility must also include all that is required to support the users of such a high-speed capability. The feasibility study required consideration of all parts of the proposed NASF system. The depth of study of each part of the system varied depending on the complexity of that part of the system, on the impact of that part on the system capabilities and on whether or not there was sufficient prior knowledge about how to implement that part of the system.

The evaluations performed as part of the study focused on three major issues. First the ability of the proposed system architecture to support the anticipated workload was evaluated. Second, the throughput of the computational engine (the Flow Model Processor) was studied using real application programs. Third, the availability, reliability, and maintainability of the system were modeled. The evaluations were based on the Baseline Systems of the Preliminary Studies [1, 2] as modified where appropriate during this study.

The results of these evaluations show that the implementation of the NASF, in the form considered, would indeed be a feasible project with an acceptable level of risk. The technology required (both hardware and software) either already exists or, in the case of a few parts, is expected to be announced this year.

This executive summary explains the results of the major areas of the study. First, the basic goals will be summarized. Then the system considered for the NASF will be described. Next the results of the three major evaluations will be explained. Finally, some of the key hardware and software concepts will be described.

STUDY OBJECTIVES

The principal objective of the study has been to consider the feasibility that a facility (NASF), which could support a throughput well in excess of what would be commercially available, could be implemented. In particular, the goal is to have a system where time-averaged Navier-Stokes computation can be performed in 10 minutes or less (on steady fluid flow problems involving a million grid points). Not only is this throughput goal important, but since the intent of the facility is to support daily usage by a large user community, the NASF system availability needs to be better than 90% and the facility needs to be nominally available for 22 hours a day. In order that the NASF may support long runs, the mean time between interruptions should be longer than ten hours. In some cases, an alternate form of the throughput goal can be used. A sustained, average rate of execution of one billion floating point operations per second (one gigaflop/sec or 1 GFLOPS) corresponds roughly to the problem throughput desired on the aerodynamic flow codes.

The starting point of the effort in this study was the baseline configuration developed during the Preliminary Study under contract NAS2-9456 [1,2]. The overall goal was to gain an understanding of the characteristics, capabilities, and potential of the facility in order to make a judgment as to its feasibility. The study required the development of further specifications in order to consider the responsiveness to the desired application of the facility and to develop estimates of the schedule, cost, and risk of such a development.

Both functional and performance (timing) simulators were developed to be able to estimate (as accurately as possible) performance and reliability of the system. Although the primary application of the facility is likely to be aerodynamic flow modeling, the performance studies included both aerodynamic flow codes and weather modeling codes. The use of real programs in these application areas allowed an initial evaluation of the flexibility of the language constructs proposed. This evaluation was especially important since the facility needs to be sufficiently flexible that algorithm development could be supported for fluid dynamics algorithms as yet not investigated. In addition, the diverse user needs for input, output, and algorithm investigation must be supported.

Since the development of the baseline systems considered only aerodynamic flow modeling applications, the consideration of weather modeling codes was especially important. This consideration was used to evaluate the flexibility of the system as far as its support of other, related application areas and was used to determine whether further improvements might be needed to support these additional applications.

All of the goals could be met by the system described as a possible NASF configuration. No hardware modifications would be needed for weather code optimization. Some minor software extensions were proposed based on the weather code evaluations.

SYSTEM DESCRIPTION

Before describing the system evaluated during this study, the importance of considering all aspects of the facility must be emphasized. During the development of the system the focus tends to be on the hardware and system software (such as operating systems and compilers). As shown in Figure 1, such a focus is limited. If only the system expense is considered, the other areas important to the successful utilization of the facility may be slighted. In particular, users themselves face both the expense of their training in the use of the system and the day to day expense of developing and using their various application programs. This usage would include algorithm development, programming, model description, data reduction, and so on. The users must be supported by a staff and whatever other support might be needed to keep the facility operational. Such support might include operators, power, cooling, training and supplies. Although the consideration of all these factors complicates the development of the facility, these factors must be carefully considered in order to have a facility that would not only be economical to acquire but also be economical to use. The system described below did consider these factors.

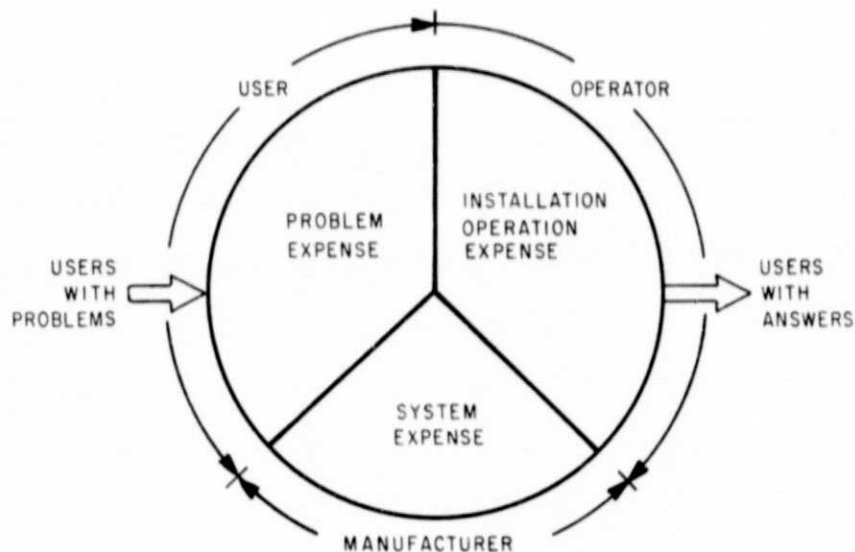


Figure 1 Total Cost of NASF Usage

HARDWARE

The system originally defined during the Preliminary Studies and modified during this study is shown conceptually in Figure 2. The Flow Model Processor (FMP), which provides the required computational power, is a dedicated computing engine with an architecture based on the special needs of modeling. The Support Processor, the Peripheral Support System and the File System together constitute the Support Processing System. The Support Processing System interfaces with the users, maintains the data files and controls the flow of jobs and data to and from the FMP. Not shown in the figure are the support elements including building, power, office space and cooling.

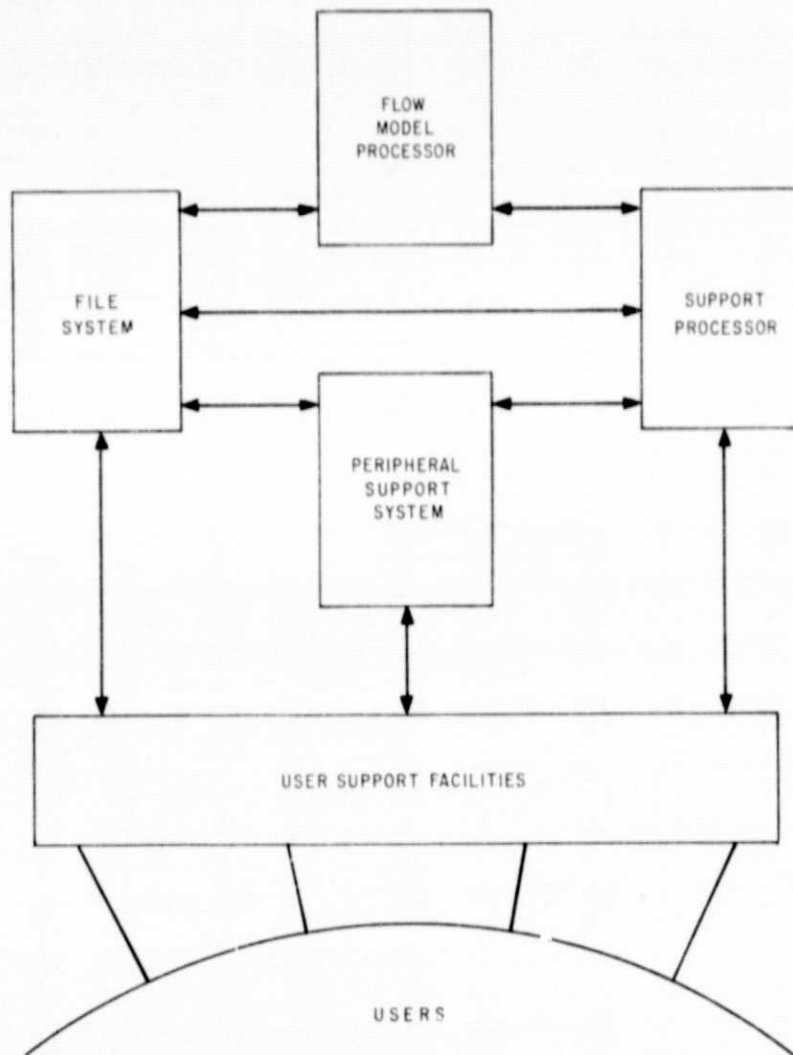


Figure 2 NASF Organization

The architecture of the Flow Model Processor is based on the needs of discrete modeling and simulation. The FMP, which is described in more detail later, has 512 processors that normally would execute independent of, and concurrent with each other. A coordinator is used to allow the processors to execute in synchronism. The processors each have memory space for programs and data. In addition, a large memory (called the Extended Memory) can be accessed by all processors through a high-speed network called the Connection Network. The Extended Memory normally would contain the data common to the processes being independently evaluated on each of the processors. Finally a slower staging memory (called Data Base Memory) would be provided to hold the next job, the last job and the current job. The Data Base Memory buffers programs and data in order to provide a smooth flow of tasks to and from the FMP. The memory sizes assumed during the study were based on the aerodynamic flow codes that are expected to be the primary application on the FMP.

The Support Processing System would consist of three portions; the Support Processor, the File System, and the Peripheral Support System. The Support Processor (the host processor) would run the main portion of the operating system (called the Master Control Program). A dual-processor B7800 was assumed for evaluation purposes. Most of the user interaction with the NASF would be through the Support Processor. The File System includes disk packs, an archival store, and the manager of the files. Data paths to and from the files would exist for the FMP, for the Support Processor, and for user support. The third element considered as part of the Support Processing System is the Peripheral Support System. The Peripheral Support System has been included because the evaluations performed in the study demonstrated that at least one of the supportive tasks involved such a level of work that a special processor for that task should be considered. In particular, the evaluations demonstrated an exceptionally heavy load can be expected to support Computer Output to Microfilm (COM). This load may be in excess of 10,000 frames of graphic information per day. The Peripheral Support System would include facilities specially designed to support such exceptional loads in order to improve the load balance across the entire facility.

SOFTWARE

Not shown in Figure 2 is the software which would be used to support users and to control the efficient usage of the resources within the facility. A dialect of FORTRAN, called FMP FORTRAN, has been proposed which has a few simple extensions to standard FORTRAN. These extensions provide application-oriented approaches to use both the independent, concurrent mode of operation. In addition, statements are included which are capable of using a large number of processors at once on a single computation. Since the Support Processor would be a commercially available processor, standard languages such as ALGOL, FORTRAN, and COBOL would be used for process definition on that processor. The File System would not be programmed by the users, but would provide high-level file management and access capabilities.

The NASF operating system (called the Master Control Program, or MCP) would reside, in part, on all elements of the system. Since the Master Control Program (MCP) would be based on existing software, the major portion would reside on the Support Processor. The portion of the MCP on the FMP would manage the flow of jobs within the FMP and would be the primary focus of confidence and diagnostic procedures within the FMP.

FAULT TOLERANCE

Since the FMP will have between 200,000 and 250,000 integrated circuits, plus other components, both hard failures and transient failures can be expected. Means for preserving the integrity of the computation in the face of such failures must be provided. The level of Large Scale Integration to be used is expected to bring forth failure modes that have not been important in the past, such as background radiation which may cause transient errors in Data Base Memory. Defense against all these possibilities must be included, and has been included in the architecture described in the Final Report[3]. Where economically feasible, mechanisms for error correction have been included such as use of single error correction, double error detection (SECDED) codes in all memories. To reduce the probability of double errors in those memories where transient failures may be expected, mechanisms to "scrub" the memory by rewriting data back into memory with the errors corrected are provided. For the various types of faults which can be detected but are not easily corrected, on-line spare processors and memory modules can be automatically switched in under control of the MCP to replace failed elements.

Not only was the FMP considered when developing the necessary fault tolerant aspects of the system. The CPU in the B7800 Support Processor is duplexed, for example, as are the Data Communications and Input Output Processors. A distributed control scheme and a multiplicity of disk packs within the File System serve to keep the system available for useful work without having each and every one of them available at any given instant. The automatic recovery procedures in the software not only support the FMP as mentioned earlier, but exist as a standard part of the MCP in the Support Processor.

NASF EVALUATION

Evaluation of the NASF considered many aspects. Three specific issues received the major attention in terms of analysis performed. These issues were an evaluation of system-level capabilities to support the general work load of the facility, an evaluation of the throughput of the FMP using real programs, and an analysis of the availability, reliability, and maintainability of the system. The general approach used for the evaluation and the results observed is described below for each of these three areas. As a result of these evaluations and the other work to date, those areas which contribute to the risks of the program were identified. These areas, which relate to the assurance of success of the program, are explained below.

SYSTEM UTILIZATION STUDIES

The evaluation of the NASF system organization showed the feasibility of the system to support the expected workloads. This evaluation was based on a hypothetical, but well thought out, workload supplied by NASA [4]. System-level models were developed and used as the basis of the implementation of system analyzer programs. The models were operationally based so that they may be easily verified by direct observation of an actual system as development might progress.

The system-level evaluation included consideration of the following:

- FMP Loading

- Support Processor CPU Loading

- Average Data Transfer Rates between Files, Users, FMP and Support Processor

- Expected number of file management actions such as file creation, deletion, and accessing.

The results of the evaluation show that the dual-processor B7800 assumed could comfortably handle the expected load with the exception of the COM support activities discussed earlier. More significantly, if projection is made to equivalent processors which are likely to be available before the implementation of the facility, such processors could handle a significant amount of the COM support load. The average data transfer rates projected by the analysis are well below the channel capacities planned. Although more analysis of peak rate requirements has yet to be performed, the projections to date are consistent with the expected results.

The FMP loading, for the workload assumed, was 20 hours per day. Table 1 shows the Support Processor loading. This table shows the number of CPU hours required per hour for the various types of processors considered. Further benchmarking would be required to verify some of the assumptions made during the study.

TABLE 1

Support Processor CPU Hours Needed/Hour
(Averaged over Day)

Processor	With COM	Without COM
Similar to B7700	14.2	1.3
Similar to B7800	9.5	.9
"Future Processor"	2.8	.2

Table 2 summarizes the average data transfer loading within the system. Averages are shown both over the day and by shift. These average rates should not be used to define peak rate capabilities.

TABLE 2
NASF Data Transfer Requirements
(with COM)

	RATE (Char/Sec)			
	Daily Average	Hourly Average		
		12M-3am	5am-5pm	5pm-12M
Support Processor - File System	29,240	83,388	16,678	35,937
Support Processor - FMP	.050	.02	.08	.02
Support Processor - Users	4,453	228	8,125	187
File System - Users	24,260	3,002	45,900	1,554
File System - FMP	163,400	294,770	210,032	73,770

Table 3 summarizes the File System control activity expected each day. The terms ACTIVE, LONGTERM and ARCHIVE in the table indicate the different types of files expected to be found in the File System. Active files are those only recently created or actively used and would be on the devices with the fastest access times. Longterm files are those which have been in the active system for up to a week with little or no use before being copied onto a slower media. Some files are saved on on-line mass storage, called the Archive in the table. These files would have an access time on the order of seconds but would still be on-line.

TABLE 3

NASF File System Control Activity per Day

FILE ACTIVITY	FILE TYPE		
	ACTIVE	LONGTERM	ARCHIVE
Files Created	2483	1127	627.3
Files Deleted	2483	1127	627.3
Files Accessed	19810	827.7	118.3
Files Replaced	1302	---	---

FLOW MODEL PROCESSOR THROUGHPUT EVALUATION

Throughput of the FMP was evaluated by measuring, in simulation and by analysis, its performance on complete programs supplied by NASA. The use of entire programs for measuring performance avoids a common pitfall in predicting the performance of new and advanced computers, namely the reliance on throughput evaluations which look only at the "hard" parts of the problems, which also are by no coincidence the parts of the problem that the advanced computer is designed to work best on.

The results of the analysis of the two aerodynamic flow codes (referred to as aero flow codes) show that the goals for throughput for aero flow applications are met. One aero flow code, identified as the "3D implicit" code was projected to execute in less than five minutes at a throughput rate of 1.01 billion floating point operations per second. The second aero flow code, identified as the "3D explicit" code was projected to execute in less than seven minutes at a throughput rate of 0.89 billion floating point operations per second. Both codes were evaluated at the nominal size expected to run on the FMP, specifically one million grid points.

The results of the analysis of the weather codes shows that the FMP, as evaluated, is optimized for the weather codes as well. NASA supplied two weather (or climate) codes. The first was a version of the Mintz-Arakawa algorithm, as developed by the Goddard Institute for Space Studies ("GISS"); the second was a spectral weather code. The same detailed analysis was applied to the GISS weather that had been applied to the aerodynamic codes. Fourteen days of simulated weather, with 20 minute time steps, in a 2.5° (latitude and longitude) model with a total of 115,334 grid points, would take 8 minutes to run on the FMP with an effective throughput rate of 0.53 billion floating point operations per second. Scrutiny of the second weather code showed that it could be expected to run with slightly higher throughput than the GISS weather, but the detailed analysis was not made.

The analysis was very thorough. All programs evaluated were dissected into code segments, each of which was internally homogeneous. The throughput was estimated for each individual code segment. From an analysis of how often each code segment was executed, the individual throughput estimates were combined into an overall execution time and throughput rate.

As a verification of the hand analysis, sections of code were input to an instruction timing simulator. The code sections chosen for simulation verified throughput rates ranging from less than 0.1 GFLOPS to more than 1.5 GFLOPS. The instruction timing simulator was based on a reasonably detailed model of a processor in the FMP. The instruction times assumed in the model correspond to what could be expected using good engineering practices and a modern circuit family such as the Fairchild 100K family of ECL circuits. The times assumed in the model for access to the common memory via the Connection Network were based on detailed analysis of the Connection Network itself. A CN simulator was developed and used to analyze various access patterns including some taken from the aero flow codes. A stochastic analyzer was used to determine the probability of success in making connections. The stochastic analyzer used probability equations for analysis. Both methods validate a transfer rate through the connection network of over one billion words per second from all processors to all memory modules.

The analysis of the various programs required preparation of FMP FORTRAN versions to be used in the analysis and as the starting point for hand-compilation onto the instruction timing simulator. The conversion from the FORTRAN code supplied to FMP FORTRAN was generally straightforward. In some cases, significant reductions in the length of the code could be made because of the application-orientation of FMP FORTRAN.

AVAILABILITY, RELIABILITY, AND MAINTAINABILITY EVALUATIONS

Several methods were used to evaluate the availability, reliability, and maintainability of the NASF. The predictions for the FMP are based upon a computer model of reliability and availability with assumptions that are derived from the military standard methods for estimating reliability. In an attempt to be as realistic as possible, field data which included failures due to system software as well as hardware was used. In addition, intermittent failure modes were modeled, where the rate of intermittents was based on field experience.

With the fault tolerance mechanisms in place, the availability forecasts are 99% for the FMP by itself and over 99% for the Support Processing System. These individual predictions combine to an NASF availability of over 98%. An estimate of 14.1 hours between interruptions of processing was also made as a result of the reliability and availability modeling. These predictions for the SPS are based on field data for the B7700, which is similar to the B7800 for reliability and availability.

PROGRAM SUCCESS ASSURANCE

To assure the success of the NASF project, one must assure success in all areas. Some areas, being dependent mainly on existing technology or existing methods, were only briefly addressed during the study. Other areas of concern, especially where the NASF and its FMP represent a break with past experience, were addressed at greater length. A discussion of some of the key points addressed is summarized below.

Although outside the scope of the study, the need for continuing commitment to the successful implementation of the NASF on both NASA's and the vendor's parts must be carefully considered. The close technical interaction that was so important to the Preliminary and Feasibility Studies must be continued. The length of time from the eventual start of design to delivery of the system is long. Project attention must be kept firmly on the job at hand. Continual changes of direction, dilution of effort, and expansion of goals could make the project seem to have a constant time-to-completion. This study has shown that a project begun now, with currently available or imminently expected technology, could deliver an operational system which would fulfill NASA's objectives.

Software development could have several potential problem areas. Software has been notoriously hard to schedule, often because of incomplete or changing specifications. Software is especially subject to the temptation to add "just one more little feature" making the resulting product more and more complex and difficult to test. This problem must be handled by careful management. The two major areas of software concern in the NASF are the operating system, and the language and compiler. The operating system (called the Master Control Program, MCP) would be based on the

existing MCP of the B7800 planned as the Support Processor. This MCP has a history of 19 years of development behind it and is already being modified by Burroughs to support job flow to the computational engine for the Burroughs Scientific Processor. With this work substantially complete, the integration of the FMP becomes a task with much less risk.

Compiler development is another area often assumed to be a problem area. Here risk has been significantly reduced by proposing a language which is essentially ANSI Standard FORTRAN with a structure surrounding the FORTRAN pieces. This structure allows the FORTRAN pieces to map directly onto the many individual processors of the FMP. The result is that most of the compilation is the same serial FORTRAN to processor-level code process that industry and Burroughs has considerable experience with. The coordination between the pieces of standard FORTRAN is simply described by the added structure and maps easily onto the section of the FMP specifically designed for such coordination (i. e., the coordinator).

As a result of the approaches proposed and evaluated during the study, the success of implementation of the necessary software seems assured.

Hardware presents no threat to the success of the project. The technology projections made during the Preliminary Study [2, 3] are proving to be conservative. Logic design would be straightforward and presents little in the way of new challenges. The organization considered is very modular which would allow implementation of the system with only a few types of modules. The one area in the hardware which represents a feature not found so far in any commercial computer is the Connection Network. This network provides the necessary data paths between the many processors and the large, common memory in the FMP. This network has been thoroughly simulated and otherwise analyzed during the course of this study.

SOFTWARE

The primary uses of the NASF are expected to be design and modeling applications. These applications can be approached either by experimentation (such as with wind tunnels) or by simulation. Figure 3 shows the relationship of these two approaches. The NASF is expected to support the abstraction of the "Real World" with some mathematical system. Mathematical conclusions will be established as a result of the simulation and these conclusions will then be interpreted to determine the desired physical conditions.

The abstraction process represents the development of algorithms to model real-world situations. The NASF should provide tools and support to assist in this abstraction process. The system considered in this Feasibility Study would provide support for the abstraction process both with simple extensions to the well-known FORTRAN language and with an interactive system which can be used to observe the results of the use of the model.

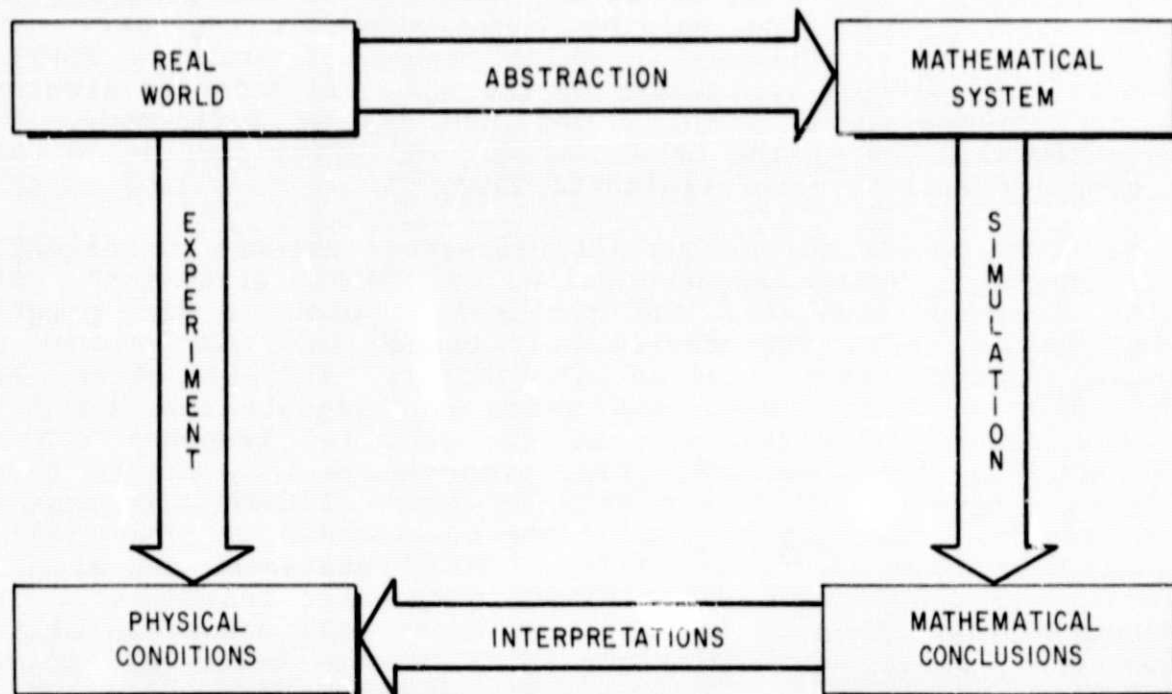


Figure 3 Relationship Between Simulation and Experimentation 13

The simulation process would also be supported with the language extensions. The Support Processing System would be used with the FMP during the simulation process to provide the same careful controls and monitoring needed during an experimentation process. The results of simulations would be observed through use of the various NASF user facilities (printers, graphics terminals, COM, etc.) for interpretation by the users. Where the results of experiments might be available on the facility, comparisons between simulations and experiments would be made.

The most direct software support of users comes from some means of describing the mathematical system which is the result of the abstraction process and of controlling the simulation process. In the NASF, the language used to define processes on the FMP provides the support required. Other forms of software support are the Master Control Program (the Operating System which controls all parts of the NASF), the File System Control Software, Intrinsics, and Test and Diagnostic Support Software.

LANGUAGE AND COMPILER

The language proposed for the FMP, called FMP FORTRAN, consists of ordinary FORTRAN segments, which do the work of the program, together with a structure holding these segments together. The structure describes the way in which pieces of ordinary FORTRAN map onto the multiple processors of the FMP. In order to simulate FMP performance, hand compilations had to be performed. The straightforwardness of the hand compilations gives confidence that the compiler will be also straightforward.

The basic mechanism for having all processors execute concurrently is a construct in the language called the "DOALL statement". The DOALL statement indicates the particular point in the program where the course of computation will branch into some number of parallel computations (called instances), all of which are independent of each other, and which can execute concurrently. For the aero flow codes, all of the parallel branches can be represented by the same code file, even though they may be doing different things on different sets of data. Figure 4 represents this parallel branching of the course of computation graphically. In the aero flow codes, the typical DOALL statement signals the branching of computation into 10,000 independent instances of the code to follow. Each of the 512 processors will handle 20 of the independent, separate computations in this case, before all 10,000 instances are completed.

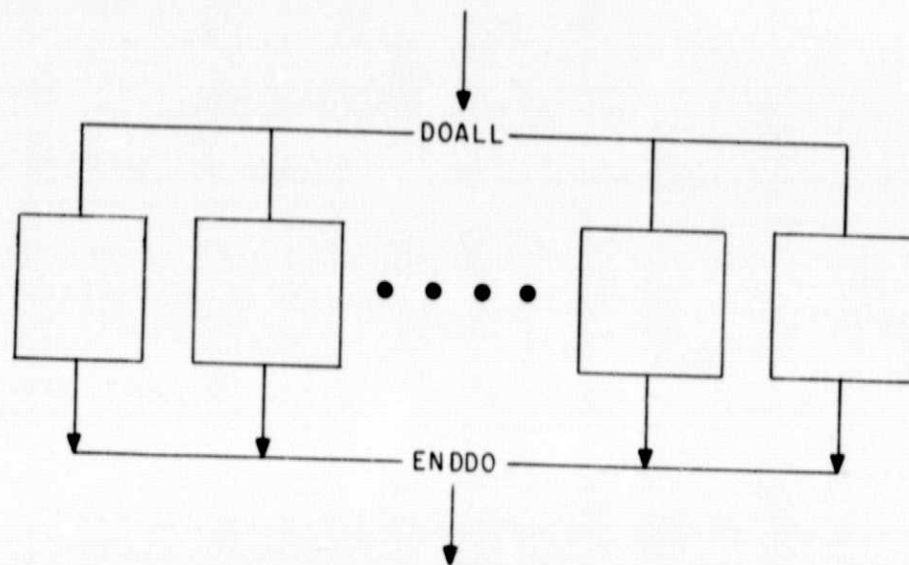


Figure 4 Specification of Concurrent Processes

OPERATING SYSTEM

The NASF should have only one operating system, pieces of which execute on the various portions of the system. In the discussions below, this operating system is called the Master Control Program (MCP). The purpose of the MCP is to provide software support for the following:

1. Scheduling and controlling the flow of programs and files to and from various processors in the system (including the support processing system and the FMP),
2. Initiating staging of jobs onto the FMP,
3. Memory management including storage management and data management,
4. Support of the FMP FORTRAN programs for functions that cannot be performed in problem mode because of overall system implications,
5. Support of other functions of the Support Processor-FMP interface such as performance monitoring, error logging and operator control,
6. Support of the external environment including interrupt handling, I/O handling, peripheral control and data communications,
7. Providing certain system utilities such as dump, and system log analyzer,
8. Support of diagnostics and maintenance for all parts of the system.

The development of a system of this magnitude is a major task. During the study of the feasibility of the NASF, the MCP considered was based on the existing MCP on Burroughs 800 series systems, in particular, the B7800. The MCP of this system has evolved from systems as early as 1960 and is, therefore, a mature system which would need no modification to satisfy many of the above requirements. Recently, Burroughs has been developing the Burroughs Scientific Processor (BSP) as an attached processor to the B7800. The general philosophies of job flow and task management in the NASF and BSP are very similar. The MCP considered is therefore based on some of the design decisions and experience gained in the BSP project.

HARDWARE DESCRIPTION

Since the Support Processing System would be standard commercial hardware, it is not discussed in detail here. The FMP, on the other hand, represents a new concept and is briefly discussed here and more completely discussed in Chapter 5 of the Final Report [3]. The Connection Network being a novel element contained within the FMP, is even more fully discussed in that report.

FLOW MODEL PROCESSOR

The architecture of the FMP draws upon past experience in many ways. Since the problem is to develop a system with an extremely high computational rate, multiprocessor systems are immediately considered.

Multiprocessor systems in which some number of processors execute code independently of each other, but share a common memory are well known. The processors may or may not have some private memory apart from the shared memory. Examples are the Burroughs' D825, B6700 and B7700, Carnegie Mellon University's Cm^x and C.mmp architecture, and others. But conventional multiprocessor systems are incapable of cooperating with each other on a short time scale, such as doing operations that require cooperation in only a single instruction-time (typically a microsecond or so). Operations such as this have been available only in the lock-step machines. Hence, the multiprocessor systems known to date cannot effectively use all the available computational power in attacking single problems. Furthermore, the number of processors in such multiprocessor systems is limited by the interconnection costs and by the communications and software overhead imposed by cooperation between processors.

Array processor systems in which some number of processing engines are all doing the same operation in lock-step with each other are also well known. One example is the Burroughs BSP. Lock-step array processor systems are limited to processing data that falls into the form of vectors. Here the arrangement of successive items of data that are to be processed simultaneously by the concurrent processors is regularly spaced in memory. Within the lock-step array processor designs there are devices which give some additional flexibility over and above the simple vector computations, but this flexibility falls far short of the flexibility which each processor in a multiprocessor array has.

When considering the expected applications, the similarity of a discrete model to multiprocessor systems is striking. However, the processes being evaluated at each of the points is, in general, a function of state throughout the model. To have a high-speed system, some means of efficient high-speed access to the "state variables" must be provided. Switching systems in which "many" (hundreds or thousands) of devices can simultaneously access many other devices are well known. Systems in which the

amount of hardware required to perform the switching has a quantity that grows as $N \log N$ (N is the number of devices), instead of growing as N^2 , have been developed. The telephone system is one example of such a system.

The architecture of the FMP has the flexibility of the multi-processor system with the efficient, high-speed access to a common storage allowed by a telephone-system-like connection. The processor power of the lock-step array type of machines is maintained for vector-oriented processing but this processing power is also available for many applications in which the data does not form vectors.

The FMP Organization is shown in detail in Figure 5. Note that there are 512 processors which can execute independent of and concurrent with each other. The various processes within a discrete model would be executed by these processors. The Extended Memory, which can be efficiently accessed from any processor via the Connection Network, is where "state variables" would be stored so as to be available to the process at any point in the model. The Coordinator provides the synchronization capability when appropriate (such as at the end of a time step). These parts of the system are described in more detail below.

The individual processors would be small, conventional, FORTRAN-oriented processors. Each is capable of executing about 3.5 million floating point operations per second, and has 32,684 words of memory private to itself.

The main memory (called "Extended Memory") is shared among all processors. It consists of 521 memory modules of 65,536 words each. The number chosen (521) eliminates memory conflicts because, being a prime number larger than the number of processors, all processors can fetch vectors in parallel when appropriate. (This concept was explained in the Preliminary Study Reports [1,2].) Thus, there is a total of 34,144,256 words in main memory plus 16,734,376 words in the processors, for a total of 50,878,632 words of program-accessible memory.

The coordinator serves as the focal point for the processor synchronization. It also executes FMP-resident system software. The coordinator is also the control interface between the FMP and Support Processing System.

A Data Base Memory (DBM) serves as the staging area for data coming in and out of the FMP. This data is transferred to and from the File System portion of the SPS. The data base memory size can easily be adjusted. The size used for the study (134,217,728 words) was picked on the basis of what was necessary for the applications considered in order to stage the next job, to hold the results of the previous job, and contain the fields necessary for the current job. A second use of the Data Base Memory is to allow the execution of jobs too large to fit into the 50 million words of main memory. The DBM could be easily expanded to larger sizes as needed.

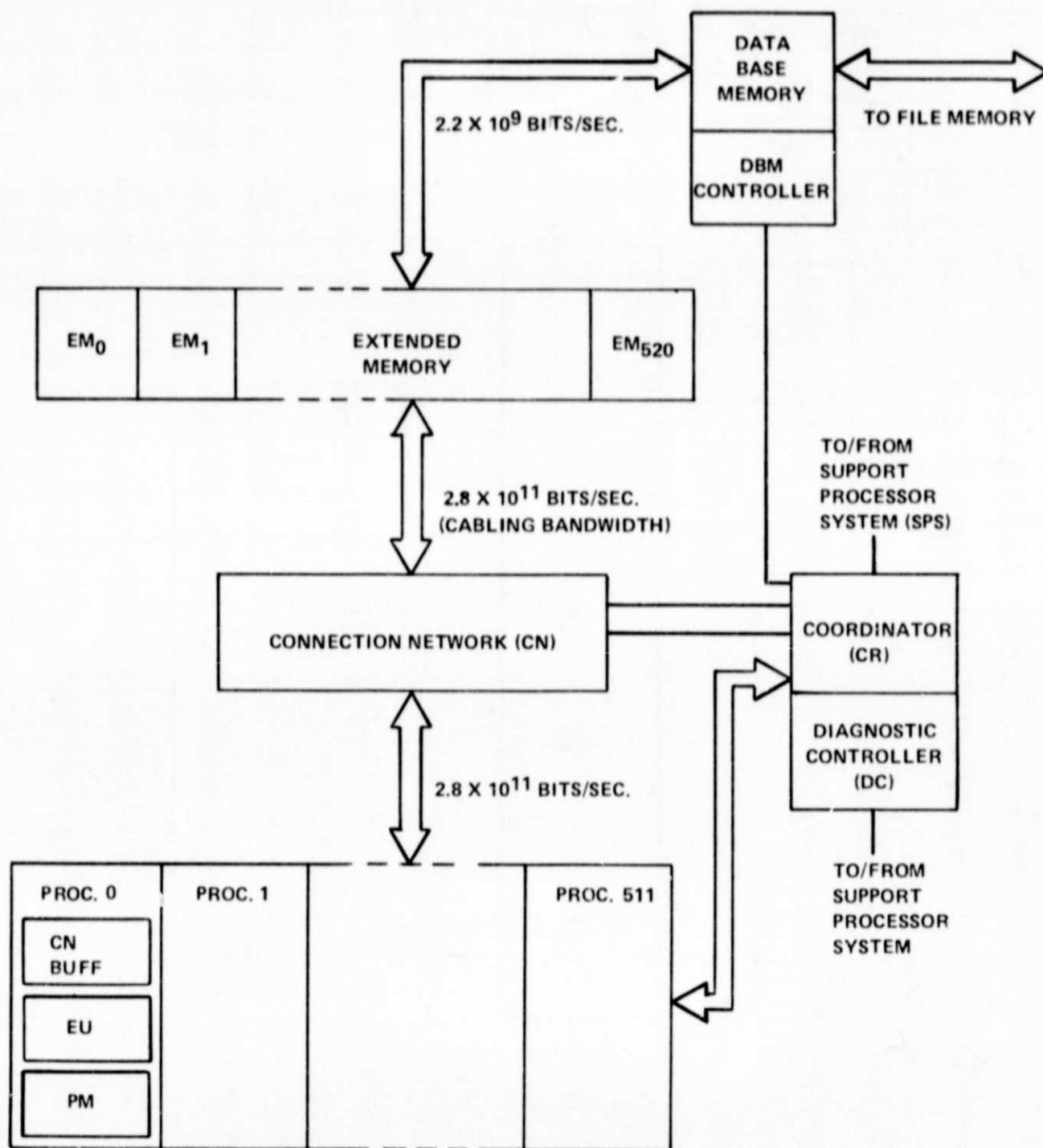


Figure 5 FMP Organization

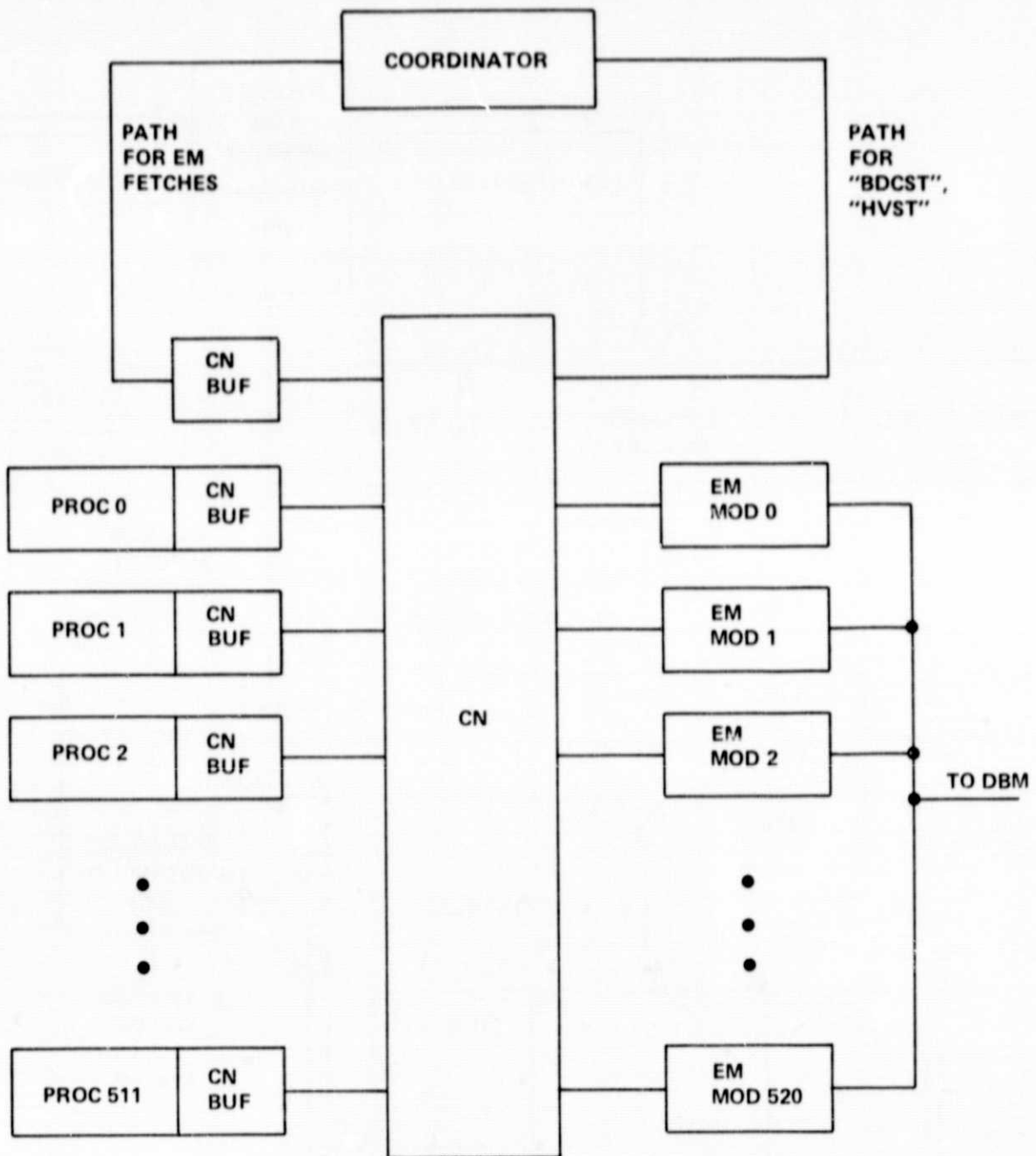


Figure 6 Connections to CN In FMP

The last major element of the FMP is the Connection Network (CN). As can be seen in Figure 6, the CN provides simultaneous connection between a large number of "ports" but has a low total parts count. No existing computer uses an arrangement similar to this one, hence a substantial design validation was carried out.

Control of diagnosis of the FMP is with the Support Processor. The Diagnostic Controller (DC) is the part of the FMP which the diagnostic controls in the Support Processor use during maintenance procedures. The DC provides diagnostic access to the Coordinator and through the Coordinator to the rest of the FMP.

TECHNOLOGY

The FMP is expected to be built of emitter-coupled logic (ECL), which is the mature high-speed logic family that has been available for many years, and in which a growing number of functions are becoming available in LSI. For example, an 8-bit slice of an arithmetic unit is now available in ECL, so that most of the arithmetic unit of a 48-bit machine consists of eight packages.

The system evaluated was based on the technology projections made during the Preliminary Study. These projections have been conservative to date. For example, the 64K-bit memory chips recently announced would be almost fast enough to be used as processor memory parts instead of the 16K-bit parts originally projected. As this example shows, at the current rate of technological advance in the semiconductor industry, it would be foolish to settle on any particular semiconductor component available today as the component choice for the FMP. In the memory area, as well as logic, this is true.

CONCLUSION

The work summarized above has demonstrated the feasibility of the Numerical Aerodynamic Simulation Facility. Although some risks have been identified, the level of risk is low for the architecture and software considered during the evaluation. This system is believed to be the best approach to meeting the total system goals for the NASF. In particular, with these concepts no new advances, beyond the technology available today, are needed in order to successfully implement the facility.

REFERENCES

- [1] Final Report, Numerical Aerodynamic Simulation Facility, Preliminary Study, October, 1977, Contract NAS2-9456, Burroughs Corporation, Paoli, PA. NASA CR 152060 152061, 152062.
- [2] Final Report, Numerical Aerodynamic Simulation Facility, Preliminary Study Extension, February 1978, Contract NAS2-9456, Burroughs Corporation, Paoli, PA. NASA CR 152106, 152107
- [3] Final Report, Feasibility Study of the Numerical Aerodynamic Simulation Facility, March 1979, Contract NAS2-9897, Burroughs Corporation, Paoli, PA
- [4] NASF UTILIZATION, October 1978 (NASA Ames), Draft as amended through subsequent communications.